AIAA 96-2827

Real-Time Flight Data Validation for Rocket Engines

R.L. Bickford
ExperTech
Orangevale, California

T.W. Bickmore
Intelligent Software Associates, Inc.
Sacramento, California

C.M. Meyer
NYMA, Inc.
Brookpark, Ohio

J.F. Zakrajsek
NASA Lewis Research Center
Cleveland, Ohio

32nd AIAA/ASME/SAE/ASEE
Joint Propulsion Conference
July 1-3, 1996 / Lake Buena Vista, FL

REAL-TIME FLIGHT DATA VALIDATION FOR ROCKET ENGINES

R.L. Bickford[1], T.W. Bickmore[2], C.M. Meyer[3], J.F. Zakrajsek[4]

[1]ExperTech
Orangevale, California
rando@ix.netcom.com

[2]Intelligent Software Associates, Inc.
Sacramento, California

[3]NYMA, Inc.
Brookpark, Ohio

[4]NASA Lewis Research Center
Cleveland, Ohio

Abstract

Real-time validation of rocket engine sensor data improves mission safety and reduces flight operations and ground test costs. NASA Lewis Research Center, in partnership with ISAI/ExperTech, has developed an innovative sensor validation system enabling real-time, automated sensor failure detection for all types of mission critical systems. Work to date has verified that these sensor validation algorithms enable highly reliable data validation for critical Space Shuttle Main Engine performance sensors, including the turbine discharge temperatures on both turbopumps. We have completed production of a prototype run-time module which has been shown to validate 22 SSME flight sensors in real-time with very high reliability.

The Sensor Validation System consists of a sensor validation network development system and a real-time kernel. The network development system provides the workstation-based tools that define the analytical redundancy relations and decision strategy used by the real-time kernel to detect sensor failures in a real-time data stream. The network development system includes an autocode generator which automatically produces the validation files used by the real-time kernel. The real-time kernel plus the autocode generated files form a run-time module which can be easily integrated with the host process. This design enables non-programmers to produce and maintain sensor validation run-time modules for any process application.

Background

Space Shuttle history illustrates the potential value of sensor validation.[1] Numerous test aborts, launch scrubs and launch delays have resulted from sensor failures. Sensor failures dominate the engine anomaly reports. Operations costs are adversely impacted with each test abort, launch scrub and anomaly investigation. This history, and the similar sensor failure histories of other launch vehicles and ground test facilities, prompted NASA Lewis Research Center (LeRC) to develop a highly reliable, cost effective approach to the real-time detection and identification of sensor failures.

Efforts to identify and develop an effective sensor validation strategy were initiated by NASA LeRC in 1990. Early studies included an evaluation of known algorithmic approaches for automating the sensor validation process.[2] From these studies, a prototype algorithm based on analytic redundancy and Bayesian belief networks was defined. This algorithm was then used to develop a prototype software module configured to validate the high pressure fuel turbine discharge temperature (HPFT DS T) sensors for the Space Shuttle Main Engine (SSME).[3] Simulation laboratory testing was followed by five hot-fire evaluation tests at the NASA Marshall Space Flight Center Technology Test Bed. All simulation laboratory sensor failures were detected with no false alarms in simulation or hot-fire testing. The results of these early tests and prototypes confirmed the potential of the sensor validation approach and provided the impetus for further development.

In 1995, NASA LeRC partnered with ISAI/ExperTech to develop a general purpose solution for the development and production of sensor validation systems. This effort focused on the development of a set of software tools which substantially automate this development and production and extended the prototype algorithm to provide coverage for all SSME operating states from engine start through shutdown. The capability and performance of this set of software tools, herein known as the Sensor Validation System (SVS), was then verified by the production of a prototype run-time system which validates 22 SSME flight sensors in real-time with very high reliability. It is this development and testing effort which forms the basis for this paper.

Requirements for sensor data validation exist in any mission critical aerospace or industrial control and safety system where data integrity is essential. The technology is applicable to any process or vehicle control application where:

- Time-critical, closed-loop control and safety monitoring depend on sensor input;
- Unexpected process interruptions due to sensor failures or false alarms are uneconomical.

The SVS automates the production of application specific sensor validation run-time modules which are embeddable in the user's process control environment. These real-time capable modules enable improved safety, reduced maintenance cost, and optimal economics for many process-oriented enterprises including aerospace vehicle and ground support systems, power generation plants, and chemical processing plants. The goals of the sensor validation system are:

- Prevent process safety system false alarms and unnecessary shutdown and maintenance;
- Ensure closed-loop control integrity to optimize process economics, extend hardware life and assure mission success;
- Automatically verify the integrity of the vehicle or plant sensing systems;
- Ensure reliable 'red-line' safety protection for personnel and equipment;
- Ensure that automated diagnostic systems 'reason' with valid data;
- Minimize the manpower, schedule and uncertainty associated with sensor failure identification and remediation.

Sensor data failures are defined as any failure in the data path which corrupts the sensor signal, thereby providing erroneous information to the process control or monitoring system. Thus, sensor validation modules are effective at identifying failures in sensors, cables and data acquisition electronics. The algorithm isolates the specific sensor which has failed and may be modified to generate a synthetic replacement signal, if necessary.

We have used the SVS to produce a real-time software module which detects SSME sensor data failures with very high reliability. This SSME prototype has been tested in workstation and embedded applications by NASA, ISAI/ExperTech, Boeing and Lockheed-Martin, all with excellent results. Many of the X-33 and RLV concepts incorporate SSME derivatives for the primary propulsion system, making our effort directly applicable to both the RLV and Space Shuttle programs. More importantly, sensor validation technology may be applied to many other flight critical subsystems, including:

- Reaction control systems;
- Propellant delivery systems;
- Auxiliary power systems;
- Inertial navigation systems;
- Crew life support systems;
- Ground support systems.

Sensor Validation Approach

Our mathematical approach combines analytic redundancy and Bayesian decision theory to provide a solution which has well-defined real-time characteristics, well-defined error rates, and is scalable to validate any number of system sensors.[4] Analytical redundancy is a technique in which a sensor's value is predicted from the values of other sensors and known or empirically derived mathematical relations. System design relationships and sensor redundancies provide many of these types of models. Relations can also be empirically derived using techniques such as statistical analysis, pattern recognition and neural networks.[5].

A set of sensors and a set of relations among them form a network of cross-checks used to periodically validate all sensors in the network. The difference between a value predicted using a relation and a directly sensed value is called a *residual.* The probability of a relation holding, given that all related sensors are valid, is determined statistically by placing a threshold value on the relation residual. Statistical properties of the relation residuals are pre-computed using nominal system operating data. Threshold values are typically set at the three standard deviation (3 sigma) value of the relation residuals based on nominal operating data.

Bayesian belief networks provide a mathematically sound method of determining whether each sensor in the network is valid, given the results of the cross-checks. Bayesian belief networks provide a rigorous approach to the problem of *information fusion* – the combination of evidence from several sources into a single, consistent model.[6] A Bayesian belief network is constructed to represent the joint probability distribution of the sensor and relation states. The probability of each sensor being valid, given the current state of all relations, can be derived from the belief network. The probability equations are used to pre-define the configuration of the validation network for real-time decision processing.

Development Tools

A set of prototype development tools has been produced and tested to verify the feasibility of automating the development, production and maintenance of real-time sensor validation modules. The prototype SVS was implemented in CLIPS, a rule-base expert system shell developed by NASA Johnson Space Center. The prototype SVS provides a menu-driven user interface and is compatible with SunOS and MS-DOS based platforms.

The SVS development tools are used for initial production and maintenance of the embeddable run-time modules which perform the real-time validation function in the user's process environment. These run-time modules are produced automatically by the SVS autocode generator. A run-time module consists of the following three components:

Run-time kernel – The general purpose embeddable sensor data validation engine which performs real-time analysis of input data for decision processing.

Validation network – The application specific sensor and model relationships which describe analytic redundancies expected in the input data.

Voting table – The application specific decision and thresholding methodology used by the run-time kernel to determine whether a sensor has failed.

In the embedded process environment, the run-time kernel (validation algorithm) samples sensor values and determines whether each relation 'holds' or does 'not hold' by thresholding on a pre-selected residual value, such as the 3 standard deviation value. Once the status of each relation in the network is determined to 'hold' or 'not hold', the validation algorithm draws a conclusion about the validity of each sensor in the network. Conclusions from several consecutive validation cycles are used to permanently disqualify a sensor.

As shown in Figure 1, the development process begins with the user's definition of the sensors to be validated and the analytic redundancies which exist between them. The user's knowledge of the interrelationships and dependencies between sensor parameters is captured through the definition of mathematical relations (models) which define the expected redundancies in the data. Once defined, the redundancy relations are statistically fit to operations data using the model-building tools to capture normal process variations and noise present in the signals. The set of sensors and the resulting set of relations among them form a network of cross-checks used to validate all sensors in the network. We call this set of relationships a sensor validation network.

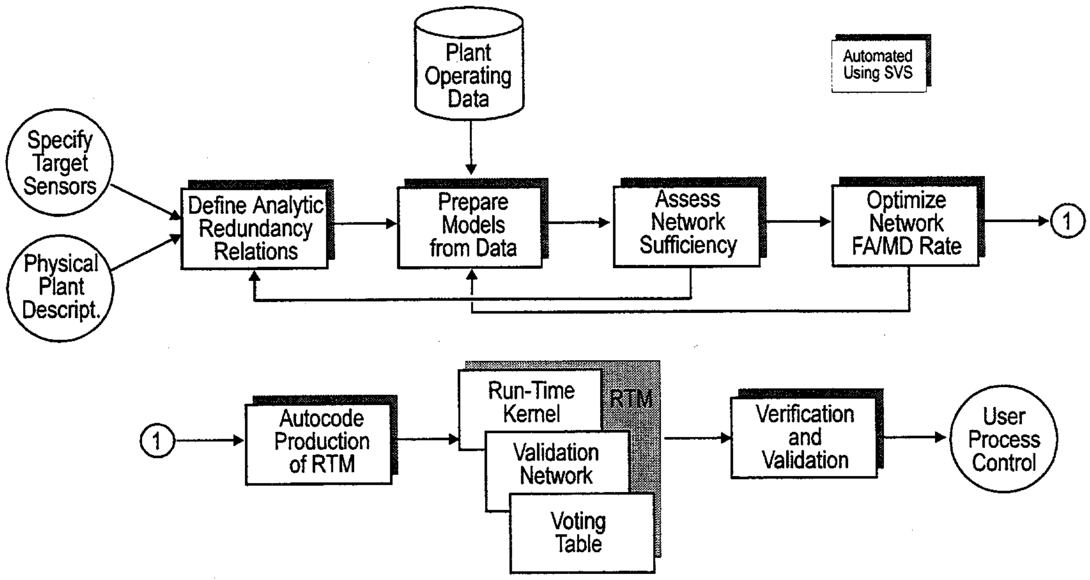
The SVS provides a framework and set of utilities to capture user knowledge of the analytic redundancies in the process data. The central purpose of the SVS is to automatically express this network of user defined relations as a Bayesian belief network and to automate the analysis and optimization of the network design and threshold parameter selection. The recursive mathematical manipulations required for network analysis are automated, thereby providing immediate feedback to the user during network development. Once an operable network is completed, the tools are used to make quantitative predictions of the system operating false alarm (FA) and missed detection (MD) rates. Quantitative error rate predictions enable the sensor validation network to be readily tuned for optimal performance prior to final production and use. In most cases, certainty in the sensor data is increased by many orders of magnitude.

SVS testing tools are provided to automate the process of testing against large numbers of actual or simulated sensor failure cases and are used to verify run-time module performance prior to field use. Single and multi-sensor failures may be simulated including hard, drift and noise failure types.

Once the user is satisfied with the network performance, the SVS provides the utilities to automatically generate the run-time module which performs the validation function in the user's process control environment. This feature enables operations and systems engineers to readily produce and maintain sensor validation code without the need for a dedicated team of software programmers.

The interface between the sensor validation run-time module and an operating system (e.g., an engine or facility controller) has been made as generic as possible. The interface consists of two simple function calls. The *initialize* function is called once to initialize the system. The *validate* function is called every controller cycle to validate the sensors. The *validate* function takes an arbitrary data structure holding the current sensor values in engineering units and a callback function which returns the current values of the requested sensors. The *validate* function also takes a second arbitrary data structure which holds the validation status for each sensor and a second callback function which performs the necessary processing to inform the host system of a sensor failure.

Figure 1. The SVS Automates a Comprehensive Methodology for Production of Embeddable Run-Time Systems



This interface makes the run-time module readily-portable, since reimplementing the two callback functions is all that is required to interface with a new operating system.

Results of SSME Application

The validation performance and real-time capability of the embeddable run-time modules have been proven by testing with actual SSME flight data. The SVS toolkit was used to design and implement a sensor validation run-time module for 22 SSME flight sensors. The SSME prototype was designed to validate 15 flight critical measurements with 7 additional measurements and 2 command parameters added to complete the analytic redundancy network. The prototype network used only binary relations (relations involving two sensors). These simple relations were shown to provide adequate redundancy for highly reliable sensor validation. Multi-parameter relations may be used in future applications to further enhance validation performance. Once the redundancy relations were selected, the analysis, optimization and production of the SSME run-time module was completed in seven days using data from 20 flights to train the network (set nominal thresholds).

The prototype SSME run-time module performed correctly on data from 50 SSME flight firings, including the proper identification of 3 sensor failures encountered in these flights. Additional development tests using simulated sensor failures demonstrated 100% accuracy on "hard" sensor failures and 100% accuracy on "soft" sensor failures (drifts and noise). A 0% false alarm rate was demonstrated in all testing. These results are consistent with the very low error rates predicted by the SVS analysis (1 error in 10^6 firings).

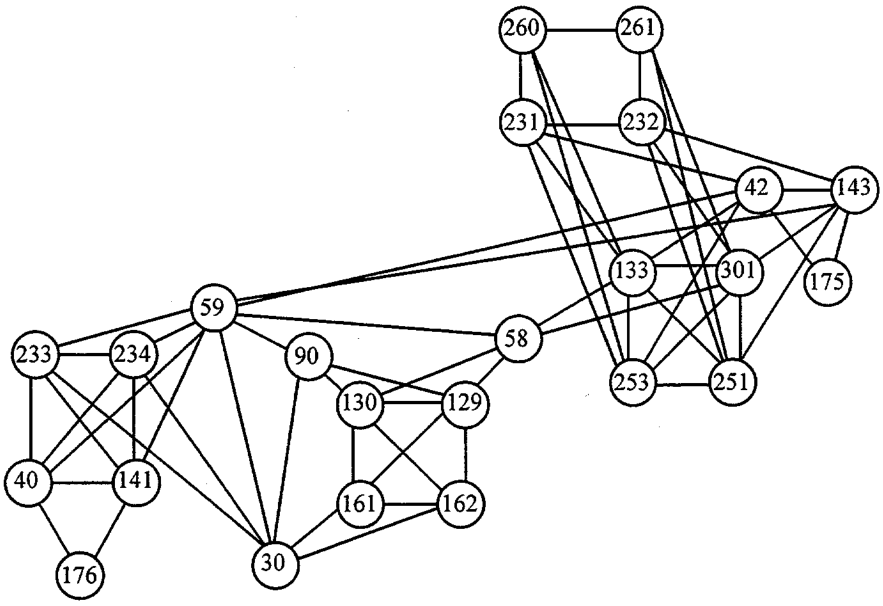
SSME Network Design

The design of the SSME prototype network, shown in Figure 2, involved a total of 22 sensors and 2 command parameters with 59 binary pairs modeled using 202 relations. Separate relations were used during each engine operating phase for the majority of the sensor pairs. The parameters included in the network are identified in Figure 3.

Characterization Testing

A series of characterization tests was performed to provide a detailed assessment of the SSME run-time module performance. SVS automated testing tools were used to perform false alarm tests on nominal data, drift tests to determine sensitivity, and simulated failure tests to determine missed detection rates. No false alarms were generated in testing against 50 flight datasets. The three sensor failures which existed in the datasets were all detected within 4 controller cycles of the onset of the event.

Figure 2. SSME Flight Sensor Validation Network Design



The results of the drift tests for steady-state engine operation are shown in Figure 4. The worst-case performance in this series occurs for HPOT DS T (PIDs 233 and 234) which is observed to drift by ~200R (~16%) before being disqualified by the sensor validation system. This is an expected result due to the high level of signal variability which is considered "normal" for this parameter. Most importantly, drift test results demonstrate a ~50% margin relative to the HPOT DS T high red-line value of 1760R and a ~65% margin relative to the low red-line value of 720R. This testing verifies that the system will disqualify a failed HPOT DS T sensor well before it reaches its flight red-line value.

Drift testing clearly demonstrates the ability of the sensor validation system to prevent safety system false alarms and unnecessary engine shutdowns. Good failure detection margins, relative to flight red-lines, were demonstrated for all red-line parameters in the network. Drift test results, expressed as a percentage of the red-line margin used prior to sensor disqualification, are summarized in Figure 5.

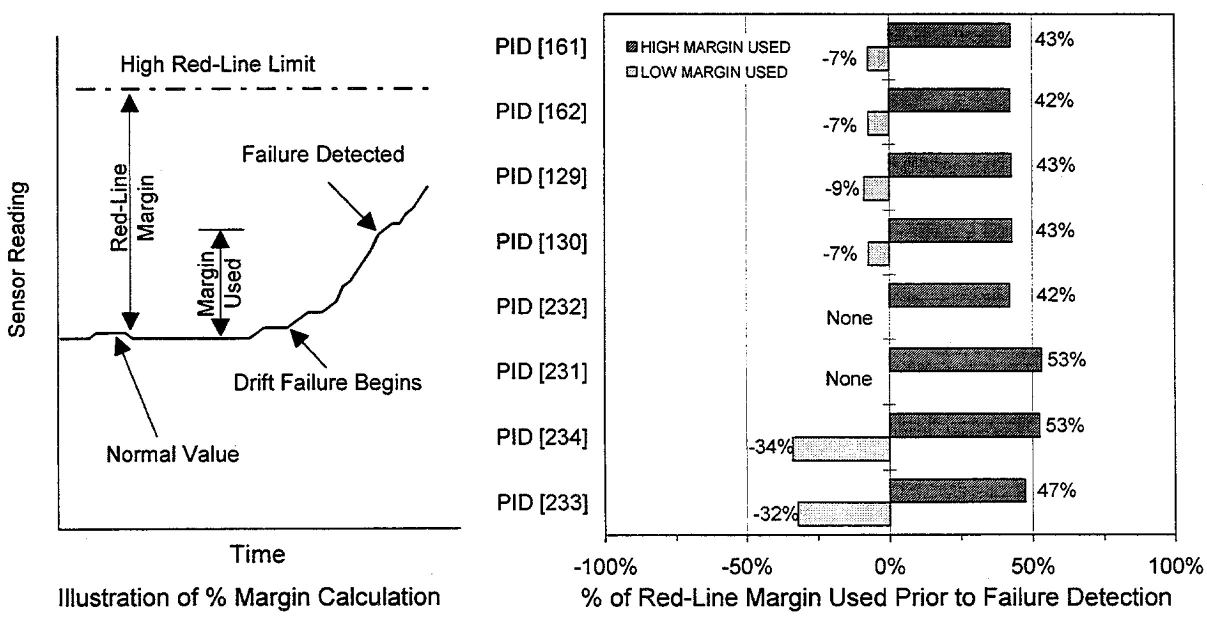
Figure 3. Flight Parameters Used In the SSME Prototype Sensor Validation System

Parameter ID	Parameter Name
30	LPOP SPEED B
40	OPOV ACT POS A
42	FPOV ACT POS A
58	FPB PC
59	PBP DS P
90	HPOP DS P
129	MCC PC A2
130	MCC PC A1
133	FUEL FLOW A1
141	OPOV ACT POS B
143	FPOV ACT POS B
161	MCC PC B2
162	MCC PC B1
175	FPOV CMD
176	OPOV CMD
231	HPFT DS T A
232	HPFT DS T B
233	HPOT DS T A
234	HPOT DS T B
251	FUEL FLOW A2
253	FUEL FLOW B2
260	HPFP SPD A
261	HPFP SPD B
301	FUEL FLOW B1

Figure 4. Steady-State Drift Test Results

Sensor	Baseline	Fail High	Fail High %	Fail Low	Fail Low %
233	1338.0	1538.0	14.9%	1139.3	14.9%
234	1347.5	1564.4	16.1%	1135.9	15.7%
231	1638.2	1809.6	10.5%	1466.7	10.5%
232	1674.0	1794.8	7.2%	1560.3	6.8%
130	3129.3	3288.9	5.1%	2975.9	4.9%
129	3127.4	3286.9	5.1%	2944.4	5.9%
162	3124.3	3283.6	5.1%	2971.1	4.9%
161	3125.4	3284.8	5.1%	2972.2	4.9%
042	80.3	86.1	7.2%	74.8	6.8%
143	79.8	84.7	6.2%	74.4	6.8%
040	65.8	74.8	13.8%	56.6	14.0%
141	65.5	70.2	7.2%	61.1	6.8%
059	7395.4	8416.7	13.8%	6424.7	13.1%
260	35122.8	36548.9	4.1%	33738.8	3.9%
261	35111.6	36537.3	4.1%	33728.1	3.9%
030	5095.0	5249.3	3.0%	4943.6	3.0%
133	15975.4	17127.8	7.2%	14890.1	6.8%
301	15968.0	16950.4	6.2%	14883.2	6.8%
251	15985.2	16968.6	6.2%	14899.3	6.8%
253	15968.0	17119.9	7.2%	14883.2	6.8%
058	5070.8	5276.6	4.1%	4870.9	3.9%
090	4072.9	4280.6	5.1%	3873.2	4.9%

Figure 5. Drift Testing Demonstrates Sensor Failure Detection Well Before Flight Red-Line Limits Are Reached



The validation system was tested against 100 randomly generated single-point sensor failures, and 25 randomly generated multiple-point sensor failures. Simulated failure magnitudes were randomly generated between 10% and 100% of the sensor full-scale range. These "failures" were translated into hard failures, drift failures, and noise failures and then overlaid on randomly selected historical data. Failure initiation times included start transient, power level transient and steady state conditions.

The results of the single-point failure tests are summarized in Figure 6. The system provided failure detection accuracy of 100% with no false alarms. Single point failure simulations included two cases during start transients, four during downthrust transients and the remainder at steady state. In three of these cases, the randomly selected data file contained real sensor failures. In each case, the system correctly identified both the real and simulated failures.

Sensor validation run-time modules are designed to dynamically reconfigure their voting strategies to accommodate multiple non-simultaneous sensor failures. Multiple sensor failure cases tested included random combinations of hard, drift and noise failure types. One upthrust transient case and 24 cases at steady state were randomly selected and tested. In one of these cases, the selected data file contained a real sensor failure. In this case, the system correctly identified and accommodated all three failures (one real and two simulated).

These results clearly verify the feasibility of using the prototype SVS and its run-time modules to reliably detect rocket engine sensor failures and eliminate safety system false alarms. Substantial automation of the network assessment, network optimization, autocode production and verification steps was accomplished and the performance of these automated tools was verified.

User Applications

Initial results have generated substantial interest in the aerospace and power generation communities and have led to cooperative demonstration activities with two aerospace prime contractors. These activities have focused on demonstrating the integration and real-time embedded performance of the SSME prototype run-time module for advanced launch vehicle applications.

Lockheed-Martin MRECS Testing

In September 1995, a cooperative demonstration effort was completed with Lockheed-Martin Space Information Systems (then Loral) to validate the real-time embedded capability of the run-time kernel and SSME prototype network. Lockheed-Martin integrated the 22 sensor SSME prototype system into their Modular Rocket Engine Control Software (MRECS) and demonstrated the sensor validation module's real-time capability in the NASA-Marshall Avionics System Testbed (MAST) laboratory. Lockheed-Martin integrated and tested the ANSI C language SSME prototype module with the Ada language MRECS software host in approximately one week.

The Lockheed-Martin testing verified the functionality of the simple, generic interface provided by the sensor validation run-time module as well as the real-time embedded processing capability of the sensor validation module in the MRECS environment Testing was performed using a static data set derived from SSME flight data.

Boeing Advanced Flight Computer Testing

In April 1996, an X-33/RLV focused cooperative demonstration effort with Boeing Defense & Space Group and NASA MSFC was completed. This effort accomplished the real-time demonstration of the SSME prototype module running embedded in Boeing's advanced fault tolerant flight computer in the NASA MAST laboratory. Boeing integrated the SSME run-time module within their Ada language flight software and performed stand-alone verification testing using SSME flight data over a period of approximately two weeks.

Figure 6. Detection Performance for Single Sensor Failure Simulations

Failure Type	Number of Tests	% Detected	Average Detection Time
HARD	36	100%	0.1 sec
DRIFT	31	100%	0.74 sec
NOISE	33	100%	1.2 sec

NASA MSFC modified facilities in the MAST laboratory to "play back" actual SSME flight sensor signals and enable true real-time testing. The capability to "overlay" failures on actual flight data was also provided by the MAST simulator. A comprehensive test series was performed which successfully demonstrated the real-time capability of the SSME run-time module to validate flight sensor data. All sensor failures were detected, including multiple non-simultaneous failures within a single test, with no false alarms generated.

The execution time of the SSME prototype was determined to be on the order of 3-msec per data cycle on Boeing's R3000 host running at 25-MHz.

Work in Progress

Development of sensor data validation methods and SVS tools is on-going with a current emphasis on:

- Enhanced discrimination between sensor and plant failures;
- Validation of high data rate sensors, such as accelerometers;
- Integration of heuristic validation methods with the Bayesian analytic redundancy method;
- Advanced statistical thresholding algorithms;
- Advanced redundancy modeling methods;
- Sensor validation module development using pre-production design and simulation data.

These extensions will improve the versatility and performance of the SVS for a wide range of sensor validation application domains.

Concluding Remarks

Efforts to date have conclusively demonstrated the feasibility of using the Sensor Validation System for rapid construction of embeddable software to reliably detect rocket engine sensor failures. The integrated set of software tools makes optimal use of the technical expertise of operations and systems engineering specialists while requiring minimal programming skills. The tool set captures the user's knowledge of analytic redundancy in the sensor data. Once these relations are defined, the tool set enables very rapid production and optimization of highly effective sensor validation networks. Automated production of embeddable run-time code is enhanced by a simple, proven interface to the host process control system. Rapid development capability and ease of host system integration have been proven by the production and embedded system testing of the 22 sensor SSME prototype run-time module.

Prototype testing has verified that the algorithms enable highly reliable data validation for mission critical SSME performance sensors, including the turbine discharge temperatures on both turbopumps. Many of the X-33 and RLV concepts incorporate SSME derivatives for the primary propulsion system, making SVS directly applicable to both the RLV and Space Shuttle programs. More importantly, this technology is applicable to many other flight critical subsystems which will benefit from automated detection of sensor failures in order to:

- Eliminate sensor failure induced false alarms and erroneous shutdowns;
- Minimize the manpower, schedule and uncertainty associated with sensor failure identification and remediation.

Requirements for this capability exist in many other aerospace and industrial control and safety systems where *data integrity* is essential. A broad range of applications in aircraft, heavy machinery, power plants, and chemical process plants are anticipated.

Program Information

This effort was sponsored by the NASA Lewis Research Center under funding provided by the Advanced Projects Program of the Office of Space Flight. The NASA COTR is Ms. June Zakrajsek.

Symbols and Acronyms

ACT	Actuator
AVG	Average
CMD	Command
DS	Discharge
FPB	Fuel preburner
FPOV	Fuel preburner oxidizer valve
HPFP	High pressure fuel pump
HPFT	High pressure fuel turbine
HPOP	High pressure oxidizer pump
HPOT	High pressure oxidizer turbine
ISAI	Intelligent Software Associates, Inc.
LeRC	Lewis Research Center
LPOP	Low pressure oxidizer pump
MCC	Main combustion chamber
MSFC	Marshall Space Flight Center
OPOV	Oxidizer preburner oxidizer valve
P	Pressure
PBP	Preburner boost pump
PC	Chamber pressure
PID	Parameter identifier
POS	Position
SSME	Space Shuttle Main Engine
T	Temperature

References

[1] Wong, K., "Space Shuttle Sensor Assessment", Vitro Corporation, Internal NASA HQ Report, April 1990.

[2] Makel, D., T. Bickmore, and W. Flaspohler, "Development of Life Prediction Capabilities for Liquid Propulsion Rocket Engines — Task 3, Sensor Data Validation and Reconstruction, Phase I: System Architecture Study Final Report", Aerojet Propulsion Division, Contract NAS 3-25883, March 1991.

[3] Bickmore, T.W., "A Probabilistic Approach to Sensor Data Validation", AIAA 92-3163, 28th Joint Propulsion Conference, Nashville, Tennessee, 1992.

[4] Bickmore, T.W., Real-Time Sensor Data Validation, NASA CR-195295, March 1994.

[5] Meyer, C.M., W.A. Maul, A.P. Dhawan, "SSME Parameter Estimation and Model Validity Using Radial Basis Function Neural Networks", Advanced Earth-to-Orbit Propulsion technology Conference, May 1994.

[6] Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, 1988.